Tutorial: Using the UML profile for MARTE to MPSoC co-design dedicated to signal processing

Imran Rafiq Quadri, Abdoulaye Gamatié, Jean-Luc Dekeyser

"Tutorial: Using the UML profile for MARTE to MPSoC co-design dedicated to signal processing"

Jean-Luc Dekeyser, Imran Rafiq Quadri, Abdoulahye Gamatié

INRIA LNE – LFL USTL Lille France

{Jean-Luc.Dekeyser, Imran.Quadri, Abdoulaye.Gamatie}@lifl.fr

Summary : This paper demonstrates the use of a model driven design flow for Multiprocessor System on chips (MPSoCs) such as those dedicated to intensive signal processing applications. The most intensive part of these applications is usually composed of systematic signal processing followed by intensive data processing. The systematic signal processing mainly consists of a chain of filters and regular processing applied on the input signals independently of the signal values. It results in a characterization of the input signals with values of interest. The intensive data processing applies irregular computations on these values of interest. Those computations may depend on the signal values. Examples of these applications are Software Radio Receiver, Sonar Beam Forming and Multimedia video codes.
Key words: SoC, UML, MDE, Model Transformations, Intensive Signal Processing, Telecommunications

1. Introduction

The computing power requirements of intensive signal processing applications such as video processing, voice recognition, telecommunications, radar or sonar are steadily increasing (several hundreds of GOPS (Giga Operations per second) for low power embedded systems in a few years). If the design productivity does not increase dramatically, the limiting factor of the growth of the semiconductor industry will not be the physical limitations due to the thinness of the fabrication process but the economy. Indeed, we ask the system design teams to build more complex systems faster, cheaper, bug free and to decrease the power consumption.

We propose to contribute to the improvement of the productivity of the electronic embedded system design teams. We structure our approach around a few key ideas:

- Promote the use of parallelism to help reduce the power consumption while improving the performance.
- Use of MDE (Model Driven Engineering) by separating the concerns in different models allowing reuse of these models.
- Propose an environment starting at the highest level of abstraction, namely the system modeling level.
- Automate code production by the use of (semi)-automatic model transformations.
- Develop simulation techniques at precise abstraction levels (functional, transactional or register transfer levels) to check the design in earlier phases of development.
- Prototype the resulting embedded systems of FPGA.
- Promote strong semantics in the application model to allow verification.
- Focus on a limited application domain, intensive signal processing applications.

All these ideas are implemented into a prototype co-design environment based on a model driven engineering approach, Gaspard2 [1],[2].

2. Gaspard2 environment

In the recent years, we have seen a productivity gap in SoC co-design, as the hardware evolution is out pacing the software development. System complexity is also increasing at an exponential rate and new methodologies are required to handle the issues related to SoC codesign.

One of the interesting solutions is to raise the design abstraction levels to reduce the system complexity. This brings us to MDE proposed by OMG. The heart of the MDE revolves around *models* which are an abstraction of reality. Each model is defined by a collection of concepts called *metamodels*. Finally using *model to model transformations*, it is possible to move from higher abstraction level models to low level technology modes in order to generate executable models (or source code).

Gaspard2 is a MDE oriented SoC co-design framework [3], which is compliant with the latest OMG industry standard, MARTE [4]. MARTE

allows modeling of real-time embedded systems: the application, architecture and the allocation between application and architecture in the UML graphical language. Gaspard2 has significantly contributed in the definition of the MARTE standard, such as the RSM package which allows the expression of repetitive structures of systems (application loops and hardware repetitions) in a compact manner. The RSM concepts were originally used in Gaspard2 to implement intensive signal processing applications.

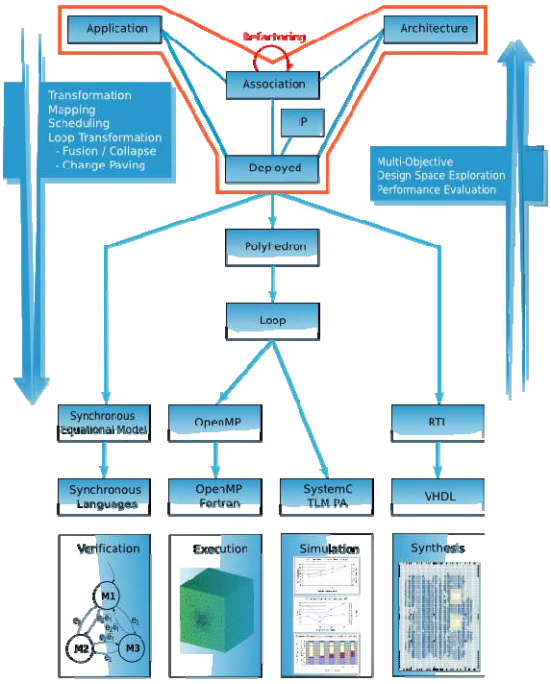

Fig. 1: Gaspard2 SoC co-design Framework

In the Gaspard2 design approach illustrated in fig.1, the application and architecture are modeled using the MARTE concepts. Afterwards, an allocation is carried out: the application part is mapped onto the available hardware resources, such as tasks on processing units and data onto memory. Although MARTE is suitable for modeling purposes, it lacks the means to move from high level modeling specifications to execution platforms. Gaspard2 bridges this gap by introducing the concept of deployment.

In the deployment level, all the elementary components or ECs, either of the application or the architecture, are linked with an implementation facilitating IP (Intellectual Property) reuse. An EC can have different implementations, which depend on the abstraction levels and execution platforms. The designer can choose an implementation among the possible choices. Hence, deployment allows one to move from platform independent models to platform dependent ones. Currently Gaspard2 targets different execution platforms such as formal verification [5], high performance computing [6], simulation [7],[8] and finally synthesis [9] as illustrated in fig.1.

3. The RTL transformation chain

This section details the way models are designed in Gaspard2 according to the above design flow. This is illustrated by considering the RTL chain which is used to synthesize hardware accelerators on FPGA. An FPGA is an electronic reconfigurable circuit which permits implementation of a complete SoC. In fig.2 we have modeled a Finite Impulse Response (FIR) digital filter. These filters are largely employed in DSP (digital signal processing) based systems and offer a large applicability range such as linear phase and stability.

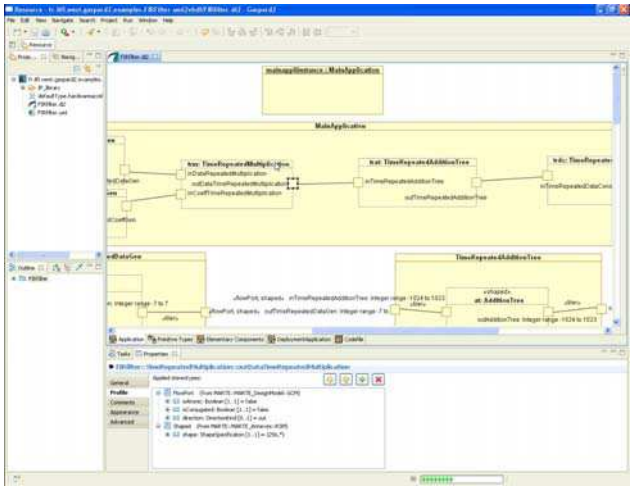

Fig. 2: Modeling of a FIR filter in our framework

The FIR filter is modeled using the MARTE concepts and the inherent task and data parallelism is clearly visible. Here the global application consists of several sub tasks. Each sub task can be either repetitive that expresses data parallelism, hierarchical that specifies task parallelism or elementary that describes an automatic function.

The type of manipulated data is mainly in the form of multidimensional arrays. The *shape* of component ports indicates the dimension of these arrays. A shape on a task component denotes the number of repetition instances associated with this task.

Once the application is modeled, it is deployed with available IPs. In fig.3, one IP is used for each EC. However, it could be possible to have several IP implementations for an EC according to designer requirements. The second part of the deployment relates to the *CodeFile*: this basically gives the information for the physical path of the source code related to an IP.

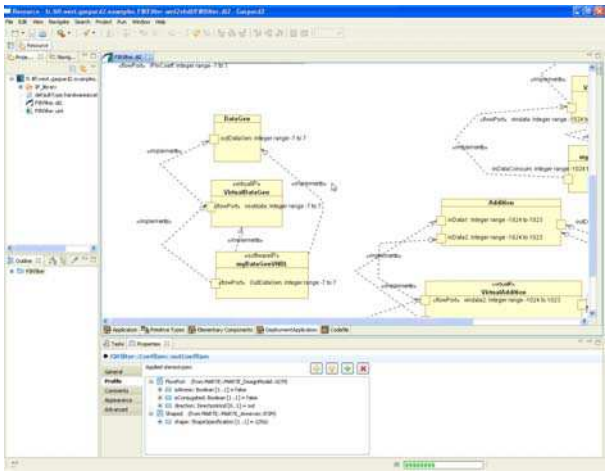

Fig. 3: Deployment of the Elementary components in the FIR filter example

Once the modeling is completed, using model to model transformations it is possible to generate the synthesizable VHDL code in order to implement the application as a hardware accelerator. Fig.4 shows a sketch of the generated source code, with each file corresponding to task of the application.

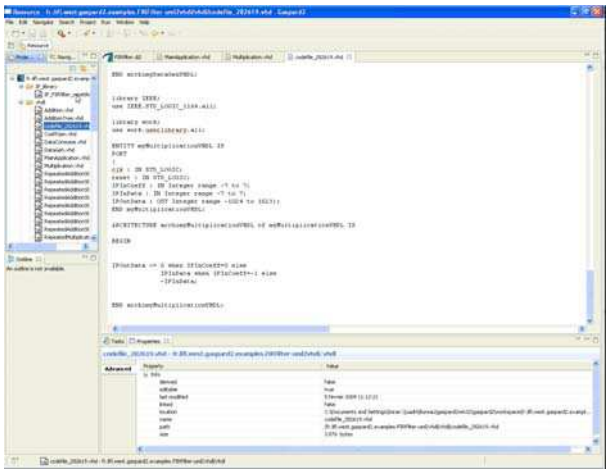

Fig. 4: Synthesizable VHDL code generated from Model Transformations

Once the code is generated, a commercial FPGA tool is used to synthesize and implement this design as shown in fig.5. Finally, a bitstream can be generated to implement the design in the targeted FPGA.

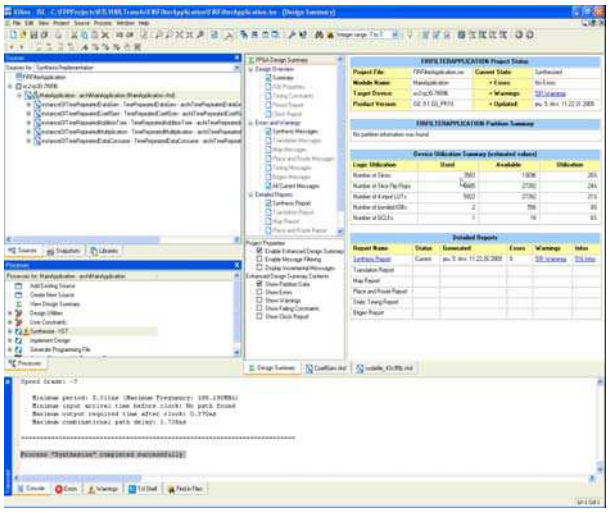

Fig. 5: Synthesis of the FIR filter on a XC2VP30 Xilinx Virtex-II Pro FPGA

4. Conclusion

The Gaspard2 platform is now compliant to the OMG standard MARTE profile, which is dedicated to the design of embedded and real-time systems. To the best of our knowledge, it is the first tool that compiles high-performance system on chip models, fully specified in MARTE. It provides designers with several compilation chains that produce code for multi-level simulation in SystemC, hardware synthesis using VHDL, high performance computing using OpenMP Fortran and C, and formal verification using synchronous data flow languages such as Lustre and Signal. This tool is available as a *Rich Client Platform* (RCP) [3].

4. References

[1] A. Gamatié et al. *A Model Driven Design Framework for High Performance Embedded Systems*. Research Report INRIA, No 6614, August 2008.

[2] DaRT Team. Activity Report 2008. available at http://www.lifl.fr/DaRT/pub/public/dart.pdf . 2009

[3] DaRT Team. *Gaspard2 SoC Framework*. http://www.gaspard2.org/. 2009

[4] OMG. *OMG MARTE Standard*. http://www.omgmarte.org/ . 2007

[5] H. Yu. *A MARTE-based Reactive Model for Data-Parallel Intensive Processing: Transformations towards the Synchronous Model*. PhD Thesis, LIFL - USTL, France, November 2008.

[6] J. Taillard. *Une approche orientée modèle pour la parallélisation d'un code de calcul éléments finis*. PhD Thesis , Lille, France, February 2009.

[7] R.B. Atitallah. *Modèles et simulation des systèmes sur puce multiprocesseurs - Estimation des performances et de la consommation d'énergie*. PhD Thesis , LIFL - USTL, France, March 2008.

[8] E. Piel. *Ordonnancement de systèmes parallèles temps-réel, De la modélisation à la mise en oeuvre par l'ingénierie dirigée par les modèles*. PhD Thesis, LIFL - USTL, France, December 2007.

[9] S. Le Beux. *Un flot de conception pour applications de traitement du signal systématique implémentées sur FPGA à base d'Ingénierie Dirigée par les Modèles*. PhD Thesis , LIFL - USTL, France, December 2007.